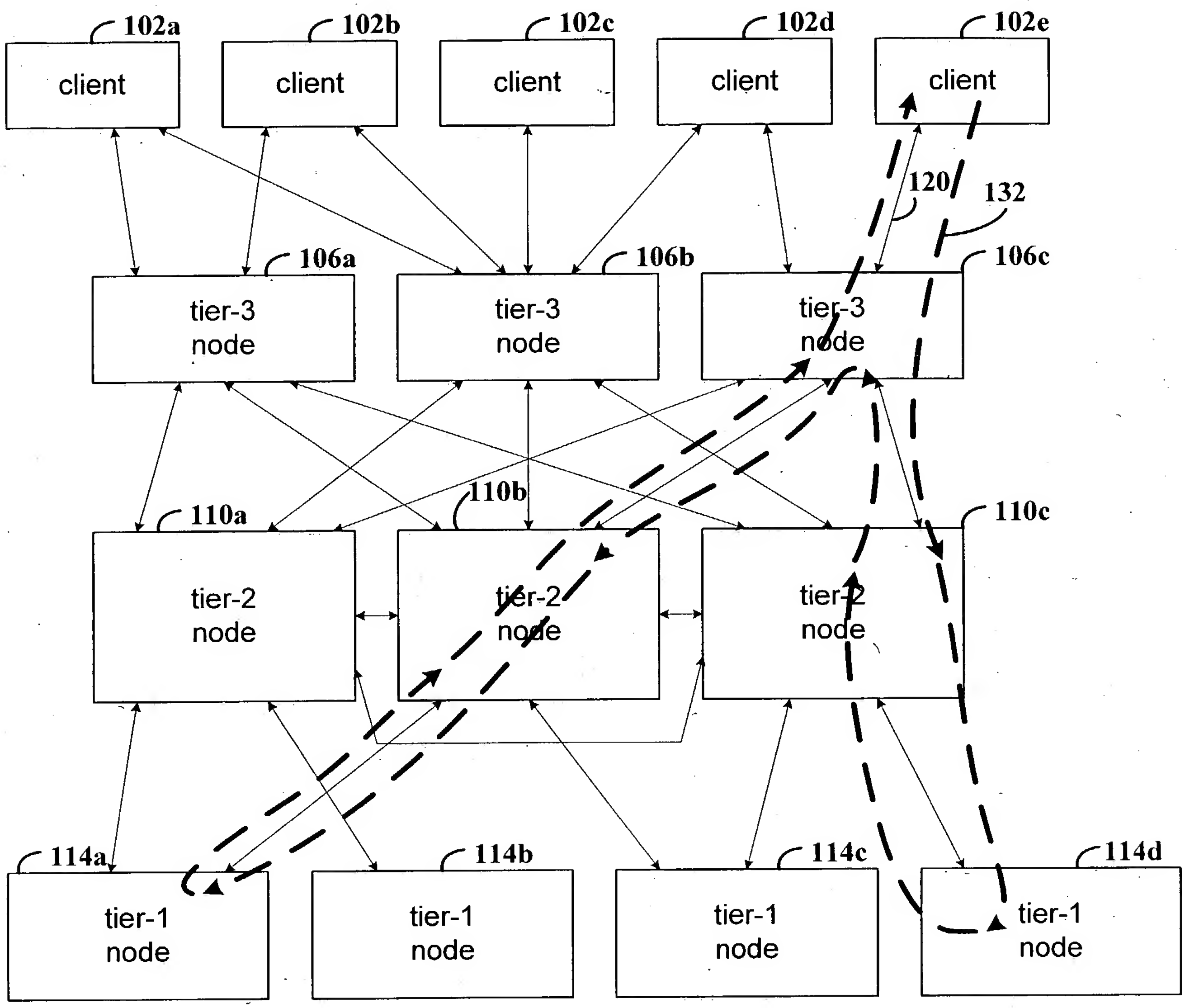
102a
102b
102c
102d
102e
client
client
client
client
client
120
132
106a
106b
106c
tier-3 node
tier-3 node
tier-3 node
110a
110b
110c
tier-2 node
tier-2 node
tier-2 node
114a
114b
114c
114d
tier-1 node
tier-1 node
tier-1 node
tier-1 node

FIG. 1

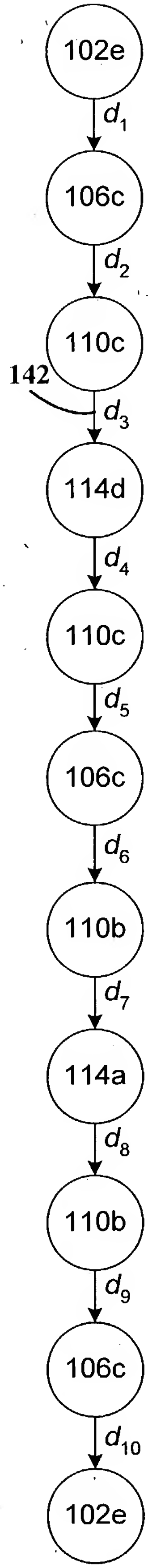

FIG. 2

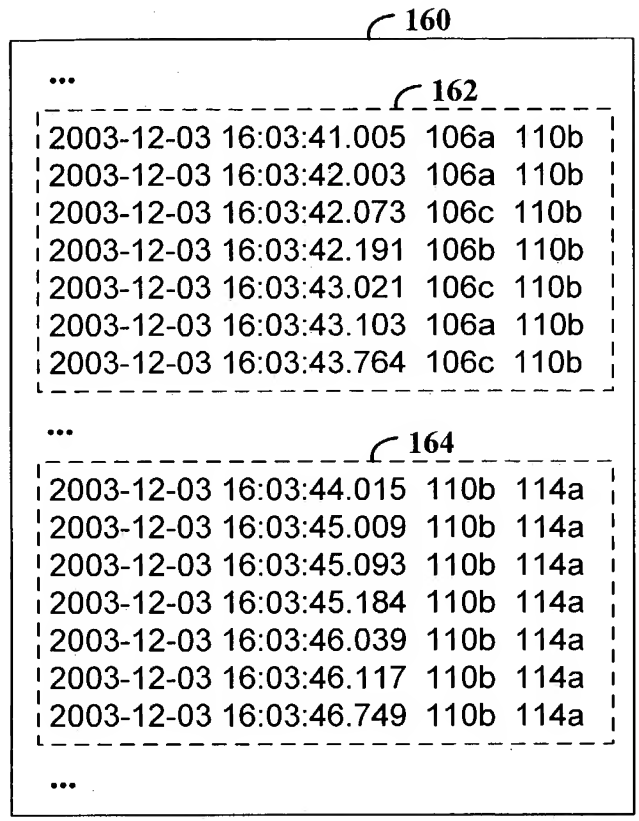

FIG. 3A

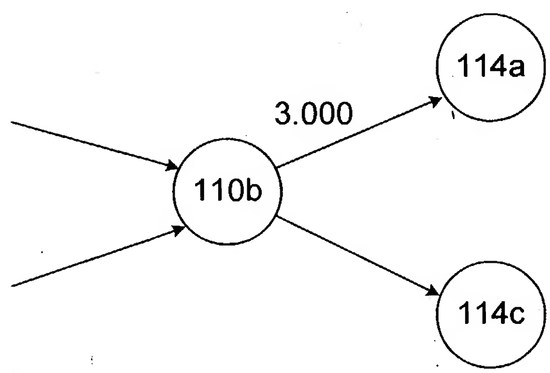

FIG. 3B

200

	GENERATE GRAPH
202	Output graph := empty
204	T_i := trace of messages with source *i*, for every *i*
206	Add a new vertex $x_{initial\_node}$ labeled initial_node to output graph
208	For each destination node *j* in $T_{initial\_node}$ with destination *j*
210	*V* := messages in $T_{initial\_node}$ with destination *j*
212	Create vertex x_j labeled *j* and edge ($x_{initial\_node}$, x_j) labeled 0 to output graph
214	Process_Node (*j*, x_j, *V*)

FIG. 4

250

	Process_Node (*j*, x_j, *V*)
252	$O_1, \ldots, O_m$:= Find_Caused_Messages (*V*, T_j)
254	For *i* := 1 to *m* do
256	*k* := O_i.node; *W* := O_i.messages; *d* := O_i.delay
258	Add a new vertex x_k labeled *k* and edge (x_j, x_k) labeled *d* to output graph
260	Process_Node (*k*, x_k, *W*)

FIG. 5

300

	Find_Caused_Messages (*V*, *Z*)
302	*i* := 0
304	*C* := Find_Correlation (*V*, *Z*)
306	Find positions of spikes of *C*(*t*)
308	For each spike position *d* found do
310	Z_0:=messages in *Z* having timestamps equal to timestamps in *V* shifted by *d*
312	For each destination node *j* in Z_0 do
314	*i* := *i* + 1
316	O_i.node :=*j*; O_i.delay := *d*; O_i.messages := messages in Z_0 with destination *j*
318	Return $O_1, O_2, \ldots, O_i$

FIG. 6

350

Find_Correlation (V, Z)	
352	$s_1(t)$:= indicator function for V
354	$s_2(t)$:= indicator function for Z
356	C := Correlation (s_2, s_1)
358	Return C

FIG. 7

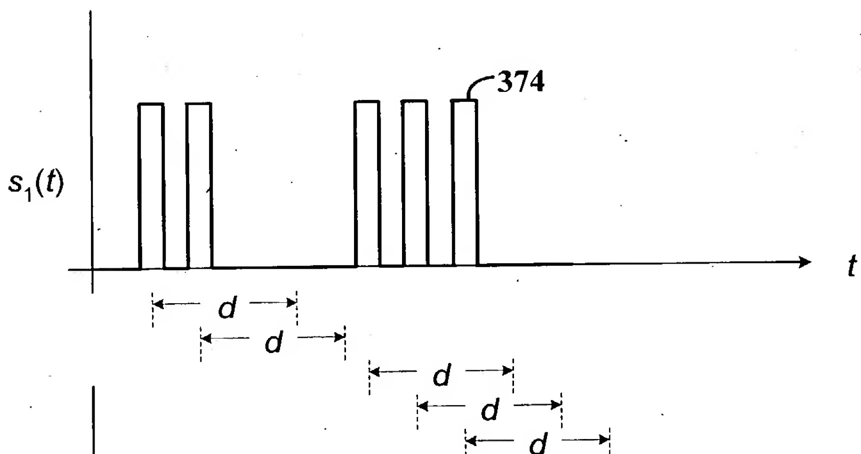

FIG. 8A

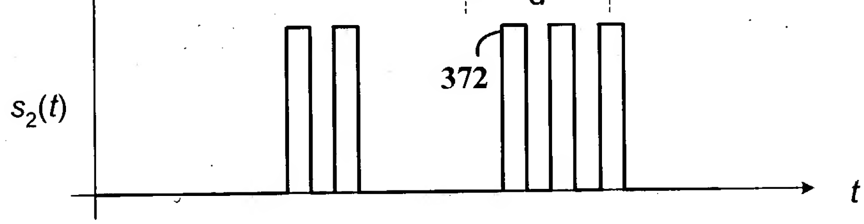

FIG. 8B

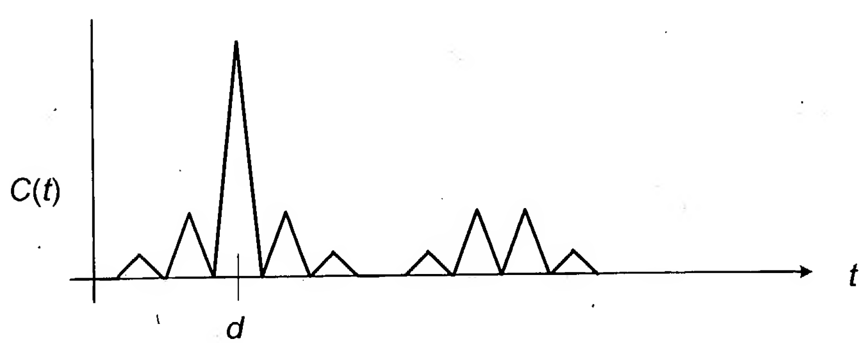

FIG. 8C

400

Find_Caused_Messages (V, Z)

402 $i := 0$

404 Nodes := Find_Related_Nodes (V, Z)

406 For each node j in Nodes do

408 Z_0 := messages in Z with destination j

410 C := Find_Correlation (V, Z_0)

412 Find positions of spikes of $C(t)$

414 For each spike position d found do

416 Z_1 := messages in Z_0 having timestamps equal to timestamps in V

418 shifted by d

420 $i := i + 1$; O_i.node := j; O_i.delay := d; O_i.messages := Z_1

422 Return O_1, O_2, …, O_i

FIG. 9

450

Find_Related_Nodes (V, Z)

452 Nodes := empty set

454 C := Find_Correlation (V, Z)

456 Find positions of spikes of $C(t)$

458 For each spike position d found do

460 Z_0:=messages in Z having timestamps equal to timestamps in V shifted by d

462 Nodes := union (Nodes, {nodes that appear as destinations in Z_0})

464 Return Nodes

FIG. 10

500

	Find_Caused_Messages (V, Z)
502	i := 0
504	Nodes := Find_Related_Nodes (V, Z)
506	For each node j in Nodes do
508	Z_0 := messages in Z with destination j
510	V_0 := V
512	While true do
514	If min $\{\lvert V_0\rvert, \lvert Z_0\rvert\}$ <= MinSize then exit while loop
516	C := Find_Correlation (V_0, Z_0)
518	If maximum of $C(t)$ is not prominent then exit while loop
520	d := position of the maximum of $C(t)$
522	Z_1 := messages in Z_0 having timestamps equal to timestamps in V_0 shifted by d
524	V_1 := messages in V_0 having timestamps equal to timestamp in Z_1 shifted by $-d$
526	i := i + 1; O_i.node := j; O_i.delay := d; O_i.messages := Z_1
528	V_0 := V_0 - V_1; Z_0 := Z_0 - Z_1
530	Return O_1, O_2, ..., O_i

FIG. 11

550

	Find_Caused_Messages (V, Z)
552	i := 0
554	Nodes := Find_Related_Nodes (V, Z)
556	For each node j in Nodes do
558	Z_0 := messages in Z with destination j
560	V_0 := V
562	W := empty set
564	Delay_set := empty set
566	While true do
568	If min {$\lvert V_0 \rvert$, $\lvert Z_0 \rvert$} <= MinSize then exit while loop
570	C := Find_Correlation (V_0, Z_0)
572	If maximum of $C(t)$ is not prominent then exit while loop
574	d := position of the maximum of $C(t)$
576	Z_1 := messages in Z_0 having timestamps equal to timestamps in V_0 shifted by d
578	V_1 := messages in V_0 having timestamps equal to timestamps in Z_1 shifted by $-d$
580	W := union (W, Z_1)
582	Delay_set := union (Delay_set, {d})
584	$V_0 := V_0 - V_1$; $Z_0 := Z_0 - Z_1$
586	$i := i + 1$; O_i.node := j; O_i.delay := Delay_set; O_i.messages := W
588	Return O_1, O_2, ..., O_i

FIG. 12

600

Find_Related_Nodes (*V*, *Z*)		
602	Nodes := empty set	
604	$V_0 := V;\ Z_0 := Z$	
606	While true do	
608		If min $\{\lvert V_0\rvert, \lvert Z_0\rvert\}$ <= MinSize then exit while loop
610		C := Find_Correlation (V_0, Z_0)
612		If maximum of $C(t)$ is not prominent then exit while loop
614		d := position of the maximum of $C(t)$
616		Z_1 := messages in Z_0 having timestamps equal to timestamps in V_0 shifted by d
618		i := node that is the most frequent destination in Z_1
620		Nodes := union (Nodes, $\{i\}$)
622		Z_2 := messages in Z_1 with destination i
624		V_2 := messages in V_0 having timestamps equal to timestamps in Z_2 shifted by $-d$
626		$V_0 := V_0 - V_2;\ Z_0 := Z_0 - Z_2$
628	Return Nodes	

FIG. 13